This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Amendments to the Claims

Claims 1 - 24 (canceled)

Claim 25 (currently amended): A method of searching and updating indexes to a data [[store]] structure in a multi-processing environment, comprising steps of:

maintaining two indexes to [[the]] a single data [[store]] structure, a first index for searching and a second index for updating;

responsive to each update of the second index, switching the indexes so that the first index becomes the second index and the updated second index becomes the first index;

allowing searches that are in progress using the first index, before the switching, to continue until completion after the switching, using ~~what is then~~ the newly-switched second index;

after the switching, initiating new searches using ~~what is then~~ the newly-switched first index;

when all searches in ~~what is then, after the switching, the~~ the newly-switched second index have completed, updating ~~what is then~~ the newly-switched second index in an identical manner as the update to which the switching step was responsive; and

preventing another operation of the switching step until completion of the step of updating the second index in the identical manner.

Claim 26 (canceled)

PAGE 4/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

Claim 27 (currently amended): A program product storage medium containing computer instructions that when executed in a computer perform a method of searching and updating indexes to a data [[store]] structure in a multi-processing environment, the method comprising steps of:

maintaining two indexes to [[the]] a single data [[store]] structure, a first index for searching and a second index for updating;

responsive to each update of the second index, switching the indexes so that the first index becomes the second index and the updated second index becomes the first index;

allowing searches that are in progress using the first index, before the switching, to continue until completion after the switching, using ~~what is then~~ the newly-switched second index;

after the switching, initiating new searches using ~~what is then~~ the newly-switched first index;

when all searches in ~~what is then, after the switching, the~~ the newly-switched second index have completed, updating ~~what is then~~ the newly-switched second index in an identical manner as the update to which the switching step was responsive; and

preventing another operation of the switching step until completion of the step of updating the second index in the identical manner.

Claim 28 (canceled)

Claim 29 (currently amended): A computer program product for serializing data structure

PAGE 5/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

retrievals and updates in a multi-processing computer system, the computer program product embodied on one or more computer-readable media and comprising:

computer-readable program code means for creating two identical data structures, [[each]] both representing an initial state for accessing a single copy of stored data;

computer-readable program code means for performing searches against a first of the two data structures, the computer-readable program code means for performing searches further comprising a first program instruction for incrementing a search use count for the first data structure atomically during each search to ensure no interference from other processes during that search and a second instruction for decrementing the search use count for the first data structure atomically after performing each search;

computer-readable program code means for performing a first update against a second of the two data structures, yielding a revised data structure;

computer-readable program code means for switching the first data structure and the revised data structure, responsive to completion of the computer-readable program code means for performing the first update, such that the first data structure becomes the second data structure and the revised data structure becomes the first data structure, the computer-readable program code means for switching the data structures further comprising a third instruction for re-ordering data structure pointers atomically to prevent interference from other processes during operation of the computer-readable program code means for switching; and

computer-readable program code means for applying, after operation of the computer-readable program code means for switching, the first update against the second data structure, yielding a second data structure that is structurally identical to the first data structure;

PAGE 6/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

the computer-readable program code means for performing searches further comprising computer-readable program code means for activating the computer-readable program code means for applying the first update against the second data structure when the search use count for the second data structure has a value indicating that no searches are being performed against the second data structure.

Claim 30 (previously presented): The computer program product according to Claim 29, further comprising:

computer-readable program code means for obtaining an exclusive lock on the second data structure prior to operation of the computer-readable program code means for performing the first update; and

computer-readable program code means for releasing the exclusive lock after operation of the computer-readable program code means for applying the first update.

Claim 31 (previously presented): The computer program product according to Claim 29, wherein the computer-readable program code means for performing the first update further comprises computer-readable program code means for queuing a transaction that specifies one or more data structure traversals and one or more data structure modifications that were performed to yield the revised data structure, and wherein the computer-readable program code means for applying the first update further comprises computer-readable program code means for performing the one or more data structure traversals and the one or more modifications specified in the queued transaction against the second data structure that results from operation of the computer-readable

PAGE 7/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

program code means for switching.

Claim 32 (previously presented): The computer program product according to Claim 29, further comprising computer-readable program code means for performing a subsequent update against the second data structure that results from operation of the computer-readable program code means for applying the first update; and wherein operation of the computer-readable program code means for performing the subsequent update causes another operation of the computer-readable program code means for switching and the computer-readable program code means for applying.

Claim 33 (currently amended): A computer system for serializing data structure retrievals and updates in a multi-processing computer system, the computer system comprising:

means for creating two identical data structures, [[each]] both representing an initial state for accessing a single copy of stored data;

means for performing searches against a first of the two data structures, the means for performing searches further comprising means for incrementing a search use count for the first data structure atomically during each search to ensure no interference from other processes during that search and means for atomically decrementing the search use count for the first data structure after performing each search;

means for performing a first update against a second of the two data structures, yielding a revised data structure;

means for switching the first data structure and the revised data structure, responsive to

PAGE 8/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

completion of the means for performing the first update, such that the first data structure becomes the second data structure and the revised data structure becomes the first data structure, the means for switching the data structures further comprising means for re-ordering data structure pointers atomically to prevent interference from other processes during operation of the means for switching; and

means for applying, after switching the data structures, the first update against the second data structure, yielding a second data structure that is structurally identical to the first data structure;

the means for performing searches further comprising means for activating the means for applying the first update against the second data structure when the search use count for the second data structure has a value indicating that no searches are being performed against the second data structure.

Claim 34 (previously presented): The system according to Claim 33, further comprising:

means for obtaining an exclusive lock on the second data structure prior to operation of the means for performing the first update; and

means for releasing the exclusive lock after operation of the means for applying the first update.

Claim 35 (previously presented): The system according to Claim 33, wherein the means for performing the first update further comprises means for queuing a transaction that specifies one or more data structure traversals and one or more data structure modifications that were

PAGE 9/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

performed to yield the revised data structure, and wherein the means for applying the first update further comprises means for performing the one or more data structure traversals and the one or more data structure modifications specified in the queued transaction against the second data structure that results from operation of the means for switching.

Claim 36 (previously presented): The system according to Claim 33, further comprising means for performing a subsequent update against the second data structure that results from operation of the means for applying the first update; and wherein operation of the means for performing the subsequent update causes another operation of the means for switching and the means for applying.

Claim 37 (currently amended): A method for serializing data structure retrievals and updates in a multi-processing computer system, comprising steps of:

creating two identical data structures, [[each]] both representing an initial state for accessing a single copy of stored data;

performing searches against a first of the two data structures, the performing searches step further comprising the step of incrementing a search use count for the first data structure atomically during each search to ensure no interference from other processes during the search and the step of decrementing the search use count for the first data structure atomically after performing each search;

performing a first update against a second of the two data structures, yielding a revised data structure;

PAGE 10/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

switching the first data structure and the revised data structure, responsive to completion of the step of performing the first update, such that the first data structure becomes the second data structure and the revised data structure becomes the first data structure, the step of switching the data structures further comprising the step of re-ordering data structure pointers atomically to prevent interference from other processes during operation of the switching step; and

applying, after the switching step, the first update against the second data structure, yielding a second data structure that is structurally identical to the first data structure;

the step of performing searches further comprising the step of activating the step of applying the first update against the second data structure when the search use count for the second data structure has a value indicating that no searches are being performed against the second data structure.

Claim 38 (previously presented): The method according to Claim 37, further comprising steps of:

obtaining an exclusive lock on the second data structure prior to performing the first update; and

releasing the exclusive lock after applying the first update.

Claim 39 (previously presented): The method according to Claim 37, wherein the step of performing the first update further comprises the step of queuing a transaction that specifies one or more data structure traversals and one or more data structure modifications that were performed to yield the revised data structure, and wherein the step of applying the first update further comprises the step of performing the one or more data structure traversals and the one or

PAGE 11/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

more data structure modifications specified in the queued transaction against the second data structure that results from operation of the switching step.

Claim 40 (previously presented): The method according to Claim 37, further comprising the step of performing a subsequent update against the second data structure that results from applying the first update; and wherein the step of performing the subsequent update causes repeating the switching step and the applying step.

Claim 41 (currently amended): A method for serializing data retrievals and updates in a computing environment, comprising steps of:

creating two identical indexes, [[each]] both representing an initial state for accessing stored data and each indexing a single copy of the stored data;

performing searches against a first of the two indexes;

performing a first update against a second of the two indexes, yielding a revised index;

serializing information describing ~~on how the first update affected the second index, including~~ a traversal path taken through [[how]] the second index ~~was traversed~~ for making the first update and one or more modifications made to [[how]] the second index ~~was modified~~ in the first update;

switching the first index and the revised index, responsive to performing the first update, such that the first index becomes the second index and the revised index becomes the first index;

applying, after the switching step, the first update ~~serialized information~~ to the second index, using the serialized information ~~about how~~ describing the traversal path and the one or

PAGE 12/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

more modifications ~~the second index was traversed and modified~~ to ~~efficiently~~ traverse and modify the newly-switched second index, thereby yielding a second index that is synchronized with, and structurally identical to, the first index; and

performing subsequent searches against the first index.

Claim 42 (currently amended): The method according to Claim 41, further comprising the step of performing a subsequent update against the second index that results from applying the ~~serialized information~~ first update; and wherein the step of performing the subsequent update causes repeating the serializing, switching, and applying steps.

Claim 43 (currently amended): A method of serializing access to data in a computing system, comprising steps of:

maintaining two trees as indexes to [[the]] a single copy of data, a first of which is used for searches and a second of which is used for update operations, each tree having a use count associated therewith;

carrying out searches using the search tree, further comprising the steps of:

determining, for each new search request, which of the trees is currently the search tree;

incrementing the use count for the search tree;

performing the new search request using the search tree; and

decrementing the use count for the search tree, responsive to completion of the performing step; and

PAGE 13/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02

carrying out each update using the update tree, further comprising the steps of:

determining which of the trees is currently the update tree;

performing an update ~~against~~ to the update tree;

serializing a record ~~of how~~ describing the update ~~affected~~ to the update tree;

switching the update tree to become the search tree and the search tree to become the update tree, responsive to completion of the steps of performing the update and serializing the record; and

applying the serialized record to the newly-switched update tree, provided that the use count for the newly-switched update tree has reached a value that indicates that no search requests are currently being performed against this newly-switched update tree, delaying the step of applying the serialized record if necessary until the use count for the newly-switched update tree has reached this value, and wherein the step of applying the serialized record ensures that both the search tree and the update tree reflect each update.

Claim 44 (previously presented): The method according to Claim 41, wherein the indexes are implemented as trees.

Claim 45 (previously presented): The method according to Claim 41, wherein the indexes are implemented as hash tables.

PAGE 14/27 * RCVD AT 6/18/2004 11:15:29 AM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/1 * DNIS:8729306 * CSID:4073437587 * DURATION (mm-ss):07-02